

Bandwidth-aware Service Placement in Community Network Clouds

Mennan Selimi
UPC
Barcelona, Spain

Llorenç Cerdà-Alabern
UPC
Barcelona, Spain

Liang Wang
Cambridge University
Cambridge, UK

Arjuna Sathiaselalan
Cambridge University
Cambridge, UK

Luís Veiga
INESC-ID Lisboa / IST
Lisbon, Portugal

Felix Freitag
UPC
Barcelona, Spain

ABSTRACT

Seamless computing and service sharing in community networks have been gaining momentum due to the emerging technology of community network micro-clouds (CNMCs). However, running services in CNMCs can face enormous challenges such as the dynamic nature of micro-clouds, limited capacity of nodes and links, asymmetric quality of wireless links for services, deployment models based on geographic singularities rather than network QoS, and etc. CNMCs have been increasingly used by network-intensive services that exchange significant amounts of data between the nodes on which they run, therefore the performance heavily relies on the available bandwidth resource in a network. This paper proposes a novel bandwidth-aware service placement algorithm which outperforms the current random placement adopted by Guifi.net. Our preliminary results show that the proposed algorithm consistently outperforms the current random placement adopted in Guifi.net by 35% regarding its bandwidth gain. More importantly, as the number of services increases, the gain tends to increase accordingly.

Keywords

Community networks, community clouds, service placement

1. INTRODUCTION

Community networks or Do-It-Yourself networks (DIYs) are bottom-up built decentralized networks, deployed and maintained by their own users. In the early 2000s, community networks (CNs) gained momentum in response to the limited options for network connectivity in rural and urban communities. One successful effort of such a network is Guifi.net¹, located in the Catalonia region of Spain. Guifi.net is defined as an open, free and neutral community network built by its members: citizens and organizations pooling their resources and coordinating efforts to build and operate a local network infrastructure. Guifi.net was launched in 2004 and till today

¹<http://guifi.net/>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

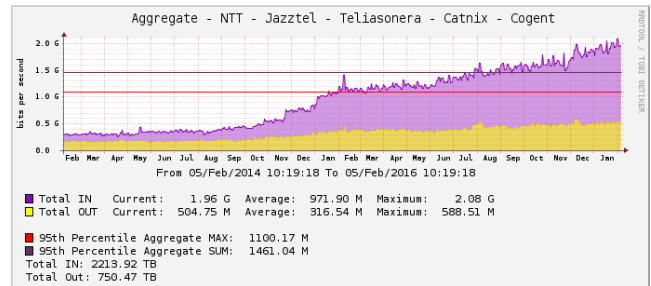


Figure 1: Guifi Traffic

it has grown into a network of more than 30.000 operational nodes, which makes it the largest community network worldwide [5]. Figure 1 shows the evolution of total inbound and outbound Guifi.net traffic to the Internet for the last two years. Pink represents incoming traffic from Internet and yellow represents outgoing traffic. For two years the traffic is doubled and peaks are as a result of a new links and fiber optics in the backbone.

Similar to other community networks, Guifi.net aims to create a highly localized digital ecosystem. However, the predominant usage we have observed, is to access the cloud-based Internet services external to a community network. For instance, more than 50% of the user-oriented services consumed in Guifi.net are gateway proxies that provide Internet connectivity hence impose a heavy burden on the limit backbone links [1]. For a very long time in the past, user-oriented services had not been developed locally because of lacking streamlined mechanisms to exploit all the available resources within a community network as well as other technological barriers. With the adoption of *community network micro-clouds*², i.e., the platform that enables cloud-based services in community networks, local user-oriented services gathered a huge momentum. Community network users started creating their own homegrown services and using alternative open source software for many of today's Internet cloud services, e.g., data storage services, interactive applications such as Voice-over-IP (VoIP), video streaming, P2P-TV, and etc. In fact, a significant amount of services were already locally deployed and running within Guifi.net including GuifiTV, Graph servers, mail servers, game servers [13]. All these services are provided by individuals, social groups, small non-profit or commercial service providers.

²<http://cloudy.community/>

Because Guifi.net nodes are geographically distributed, given this set of local services, we need to decide where these services should be placed in a network. Obviously, without taking into account the underlying network resources, a service may suffer from poor performance, e.g., by sending up large amounts of data across slow wireless links while faster and more reliable links remain underutilized. Therefore, the key challenge in community network micro-clouds is to determine the location, i.e., servers at certain geographic points in the network, where the different services multiplexed on a shared infrastructure will be running. While conceptually straightforward, it is challenging to calculate an optimal decision due to the dynamic nature of community networks and usage patterns. In this work we aim to address the following question: *"Given a community network cloud infrastructure, what is an effective and low-complexity service placement solution that maximises end-to-end performance (e.g., bandwidth)?"* Our preliminary results show that the proposed algorithm consistently outperforms the current random placement adopted in Guifi.net by 35% regarding its bandwidth gain. More importantly, as the number of services increases, the gain tends to increase accordingly.

2. NEED FOR LOCALIZED SERVICES

In this section, we characterize wireless community networks by presenting our experimental measurements in a production example over five months, which exposes the necessity of deploying localized services [18] and justifies our motivation of proposing an intelligent placement algorithm.

2.1 QMP Network: A Brief Background

The network we consider, began deployment in 2009 in a quarter of the city of Barcelona, Spain, called Sants, as part of the *Quick Mesh Project*³ (QMP). In 2012, nodes from *Universitat Politècnica de Catalunya* (UPC) joined the network, supported by the EU CONFINE⁴ project. We shall refer to this network as *QMPSU* (from Quick Mesh Project at Sants-UPC). QMPSU is part of the Guifi community network which has more than 30,000 operational nodes. At the time of writing, QMPSU has around 61 nodes, 16 at UPC and 45 at Sants. There are two gateways, one in UPC Campus and another in Sants, that connect QMPSU to the rest of Guifi.net (see Figure 2). A detailed description of QMPSU can be found in [6], and a live monitoring page updated hourly is available in the Internet⁵.

Typically, QMPSU users have an outdoor router (OR) with a Wi-Fi interface on the roof, connected through Ethernet to an indoor AP (access point) as a premises network. The most common OR in QMPSU is the NanoStation M5, which integrates a sectorial antenna with a router furnished with a wireless 802.11an interface. Some strategic locations have several NanoStations, that provide larger coverage. In addition, some links of several kilometers are set up with parabolic antennas (NanoBridges). ORs in QMPSU are flashed with the Linux distribution which was developed inside the QMP project which is a branch of OpenWRT⁶ and uses BMX6 as the mesh routing protocol [12].

2.2 Characterization: Bandwidth-Hungry

In the following, we characterize the network performance of QMP network. Our goal is to determine the key features of the network and its nodes; in particular to understand the network metrics

that could help us to design new heuristic frameworks for intelligent service placement in community networks [11]. Measurements have been obtained by connecting via SSH to each QMPSU OR and running basic system commands available in the QMP distribution. This method has the advantage that no changes or additional software need to be installed in the nodes. Live measurements have been taken hourly over the last 5 months, starting from October 2015 to February 2016. We use this data to analyse main aspects of QMP network.

Figure 3 shows the node and link presence. We define presence as the percentage a given node or link is observed over the captures. Overall, 90 different nodes were detected. From those, only 61 were alive during the all measurement period, leading to a presence higher than 98%. Around 30 nodes were missed in majority of the captures (i.e., presence less than 10%). These are temporarily working nodes from other mesh networks and laboratory devices used for various experiments. Figure 3 also reveals that 56% of links used between nodes are unidirectional and others are bidirectional.

Figure 4, depicts the Empirical Cumulative Distribution Function (ECDF) of the average traffic sent in each of the links in the busy hour. The overall average traffic observed is 70 kbps. Figure 5 shows the average traffic in both directions (upload/download) of the three busiest links.

We characterize the wireless links of the QMP network by studying their throughput. Figure 6 shows the ECDF of the throughput of the links. The figure shows that the link throughput can be fitted with an exponential distribution with mean 21.8 Mbps. In order to see the variability of the throughput, Figure 7 shows the throughput averages in both directions (upload and download) of the three busiest links (same links as in Figure 5). When we compare the Figure 7 and Figure 5, we observe that the throughput is slightly affected by the traffic in the links. Solid and dashed lines are used to identify the measurements on each direction of the links (dashed line for download, solid line for upload). It is interesting to note that the asymmetry of the throughputs measured in both directions is not always due to the asymmetry of the user traffic. For instance (node GSgranVia255), around 6am, when the user traffic is the lowest and equal in both directions, the asymmetry of the links throughputs observed in Figure 4 remains the same. We thus conclude that this asymmetry must be due to the link characteristics, as level of interferences present at each end, or different transmission powers.

A significant amount of applications that run on Guifi.net and QMP network are network-intensive (bandwidth and delay sensitive), transferring large amounts of data between the network nodes [13]. The performance of such kind of applications depends not just on computational and disk resources but also on the network bandwidth between the nodes on which they are deployed. Therefore, the placement of such services in the network is of high importance. Here are some observations (features) that we captured from the measurements in QMP network:

- QMP network is highly dynamic and diverse due to many reasons, e.g., its community nature in an urban area; its decentralised organic growth with extensive diversity in the technological choices for hardware, wireless media, link protocols, channels, routing protocols etc.; its mesh nature in the network etc. The current network deployment model is based on geographic singularities rather than QoS. The network is not scale-free. The topology is organic and different for conventional ISP network.
- The resources are not uniformly distributed in the network. Wireless links are with asymmetric quality for services (30%

³<http://qmp.cat/Home>

⁴<https://confine-project.eu/>

⁵<http://dsg.ac.upc.edu/qmpsu/index.php>

⁶<https://openwrt.org/>

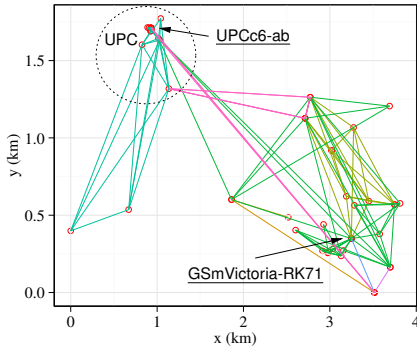


Figure 2: QMPSU network. Two main gateways are underlined.

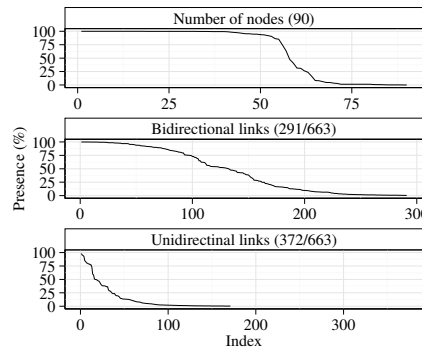


Figure 3: Nodes and links presence.

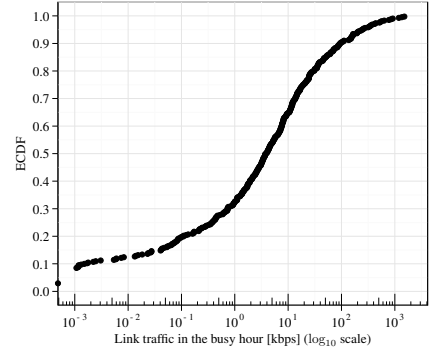


Figure 4: Link traffic in the busy hour ECDF.

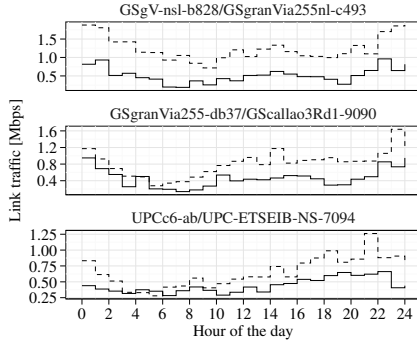


Figure 5: Traffic in the 3 busiest links.

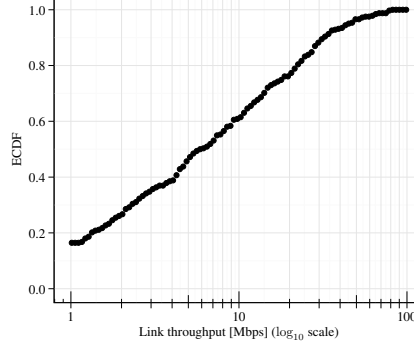


Figure 6: Throughput ECDF.

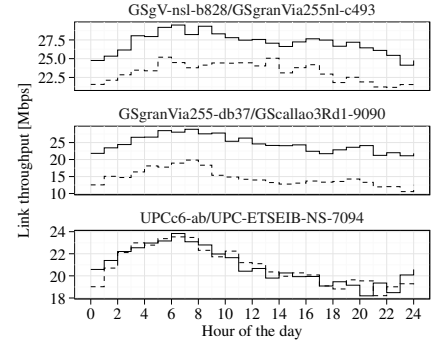


Figure 7: Throughput in the 3 busiest links.

of the links have a deviation higher than 30%). We observed a highly skewed traffic pattern (Figure 4) and highly skewed bandwidth distribution (Figure 6).

Currently used *organic (random) placement scheme* in Guifi.net community network is not sufficient to capture the dynamics of the network and therefore it fails to deliver the satisfying QoS. The strong assumption under random service placement, i.e., uniform distribution of resources, does not hold in such environments. Furthermore, the services deployed have different QoS requirements. Services that require intensive inter-component communication (e.g. streaming service), can perform better if the replicas (service components) are placed close to each other in high capacity links [14]. On other side, bandwidth-intensive services (e.g., distributed storage, video-on-demand) can perform much better if their replicas are as close as possible to their final users (e.g., overall reduction of bandwidth for service provisioning). Our goal is to build on this insight and design a network-aware service placement algorithm that will improve the service quality and network performance by optimizing the usage of scarce resources in community networks such as bandwidth.

3. BANDWIDTH-AWARE PLACEMENT

The deployment and sharing of services in community networks is made available through *community network micro-clouds* (CNMCs). The idea of CNMC is to place the cloud closer to community end-users, so users can have fast and reliable access to the service. To reach its full potential, a CNMC needs to be carefully deployed in order to utilize the available bandwidth resources.

3.1 Assumptions

In a CNMC, a server or low-power device is directly connected to the wireless base-station providing cloud services to users that are either within a reasonable distance or directly connected to base-station. These nodes are core-graph nodes what we call in Guifi.net. It is important to remark that the services aimed in this work are at infrastructure level (IaaS), as cloud services in current dedicated datacenters (we assume QMP nodes are core-graph nodes). Therefore the services are deployed directly over the core resources of the network (nodes in the core-graph) and accessed by base-graph clients. Services can be deployed by Guifi.net users or administrators.

The services we consider can be centralized or distributed. The distributed services can be composite services (non-monolithic) built from simpler parts, e.g., video streaming. These parts or components of service would create an overlay and interact with each other to offer more complex services. A service may or may not be tied to a specific node of the network. Each nodes can host one or more services.

In this work we assume offline service placement approach where single or a set of application are placed "in one shot" in the underlying physical network. We might rearrange the placement of the same service over the time because of the service performance fluctuation (e.g. weather conditions, node availability, changes in use pattern, and etc.). We do not consider real-time service migration.

3.2 Formulation and Notations

We call the community network the *underlay* to distinguish it from the *overlay* network which is built by the services. The underlay network is supposed to be connected and we assume each node knows whether other nodes can be reached (i.e., next hop is

Algorithm 1 Bandwidth-aware Service Placement (BASP)

Require: $G(V_n, E_n)$ \triangleright Network graph
 $S \leftarrow S_1, S_2, S_3, \dots, S_k$ $\triangleright k$ partition of clusters
 bw_i \triangleright bandwidth of node i

```
1: procedure PERFORMKMEANS( $G, k$ )
2:   return  $S$ 
3: end procedure
4: procedure FINDCLUSTERHEADS( $S$ )
5:    $clusterHeads \leftarrow list()$ 
6:   for all  $k \in S$  do
7:     for all  $i \in S_k$  do
8:        $bw_i \leftarrow 0$ 
9:       for all  $j \in setdiff(S, i)$  do
10:         $bw_i \leftarrow bw_i + estimate\_route\_bandw(G, i, j)$ 
11:       end for
12:        $clusterHeads \leftarrow \max bw_i$ 
13:     end for
14:   end for
15:   return  $clusterHeads$ 
16: end procedure
17: procedure RECOMPUTECLUSTERS( $clusterHeads, G$ )
18:    $St \leftarrow list()$ 
19:   for all  $i \in clusterHeads$  do
20:      $cluster_i \leftarrow list()$ 
21:     for all  $j \in setdiff(G, i)$  do
22:        $bw_j \leftarrow estimate\_route\_bandw(G, j, i)$ 
23:       if  $bw_j$  is best from other nodes  $j$  then
24:          $cluster_i \leftarrow j$ 
25:       end if
26:      $St \leftarrow cluster_i$ 
27:   end for
28: end for
29:   return  $St$ 
30: end procedure
```

known). We can model the underlay graph as: $G \leftarrow (OR, L)$ where OR is the set of outdoor routers present in the CNs and L is the set of wireless links that connects them.

Let f_{ij} be the bandwidth of the path to go from node i to node j . We want a partition of k clusters: $S \leftarrow S_1, S_2, S_3, \dots, S_k$ of the set of nodes in the mesh network. The cluster head i of cluster S_i is the location of the node where the service will be deployed. The partition maximizing the bandwidth from the cluster head to the other nodes in the cluster is given by:

$$\arg \max_S \sum_{i=1}^k \sum_{j \in S_i} f_{ij} \quad (1)$$

3.3 Proposed Algorithm: BASP

We designed a bandwidth-aware algorithm that allocated services taking into account the bandwidth of the network. We take a network snapshot (capture) from QMP network regarding the bandwidth of the links⁷. Our bandwidth-aware service placement algorithm BASP (see Algorithm 1) runs in three phases.

(i) Initially, we use the naive k-means partitioning algorithm in order to group nodes based on their geo-location. The idea is to get back clusters of locations that are close to each other. The k-means algorithm forms clusters of nodes based on the Euclidean

distances between them, where the distance metrics in our case are the geographical coordinates of the nodes. In traditional k-means algorithm, first, k out of n nodes are randomly selected as the cluster heads (centroids). Each of the remaining nodes decides its cluster head nearest to it according to the Euclidean distance. After each of the nodes in the network is assigned to one of k clusters, the centroid of each cluster is re-calculated. Grouping nodes based on geo-location is in line with how Guifi.net is organized. The nodes in Guifi.net are organized into a tree hierarchy of *zones* [7]. A zone can represent nodes from a neighborhood or a city. Each zone can be further divided in child zones that cover smaller geographical areas where nodes are close to each other. From the service perspective we consider placements inside a particular zone.

(ii) The second phase of the algorithm it is based on the concept of finding the cluster head maximizing the bandwidth between the head and member nodes of the cluster, formed in the first phase of the algorithm. The cluster heads computed in this phase are the ones having the maximum bandwidth to the other nodes in the cluster S_k . The cluster heads are node candidates for service placement.

(iii) The third and last phase of the algorithm includes reassigning the nodes to the selected cluster heads having the maximum bandwidth.

Regarding computational complexity, the naive brute force method can be estimated by calculating the *Stirling number of the second kind* [2] which counts the number of ways to partition a set of n elements into k nonempty subsets, i.e., $\frac{1}{k!} \sum_{j=0}^k (-1)^{j-k} \binom{n}{j} j^n \Rightarrow \mathcal{O}(n^k k^n)$. However, for BASP, finding the optimal solution to the k-means clustering problem if k and d (the dimension) are fixed (e.g., in our case $n = 54$, and $d = 2$), the problem can be exactly solved in time $\mathcal{O}(n^{dk+1} \log n)$, where n is the number of entities to be clustered. The complexity for computing the cluster heads in phase two is $\mathcal{O}(n^2)$, and $\mathcal{O}(n)$ for the reassigning the clusters in phase three. Therefore, the overall complexity of BASP is $\mathcal{O}(n^{2k+1} \log n)$, which is significantly smaller than the brute force method.

4. PRELIMINARY EVALUATION

Solving the problem stated in Equation 1 in brute force for any number of N and k is NP-hard. For this reason we came out with our heuristic. Initially we used k-means algorithm for a first selection of the clusters. Then, we limit the choice of the cluster heads to be inside the sets of clusters obtained using k-means. Inside these clusters we computed the cluster heads having the maximum bandwidth to the other nodes. To emphasise the importance of phase two and three, in this section we compare *BASP* to *Naive K-Means* which partitions the nodes into k groups such that the sum of squares from nodes to the assigned cluster heads (centroids) is minimized. At the minimum, all cluster heads in *Naive K-Means* are at the mean of their Voronoi sets (the set of nodes which are nearest to the cluster heads).

Our experiment is comprised of 5 runs and the presented results are averaged over all the successful runs. Each run consists of 15 repetitions. Figure 8 depicts the average bandwidth to the cluster heads obtained with *Naive K-Means* algorithm and our *BASP* algorithm. Figure reveals that for any number of k , our *BASP* algorithm outperforms the *Naive K-Means* algorithm. For $k=2$ the average bandwidth to the cluster head is increased from 18.3 Mbps (obtained with naive k-means) to 27.7 Mbps (obtained with our *BASP* algorithm) i.e., 40% increase. The biggest increase of 50% is when $k=7$. Based on the observations from the Figure 8, the gap between two algorithms is growing as k increases. K increases as network grows.

Note that our heuristics enables us to select nodes (cluster heads)

⁷<http://tomir.ac.upc.edu/qmpsu/index.php?cap=56d07684>

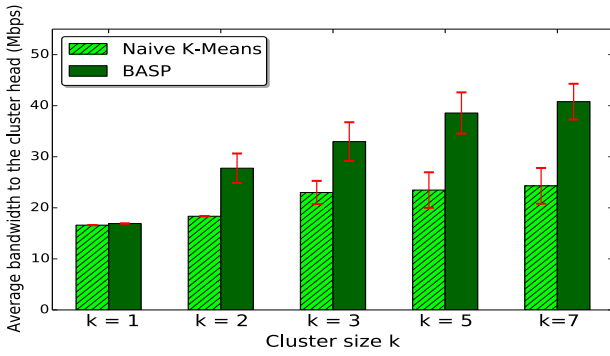


Figure 8: Average bandwidth to the cluster heads

that provide much higher bandwidth than any other random or naive approach. But, if we were about to look for the optimum bandwidth within the clusters (i.e., optimum average bandwidth for the cluster), then this problem would end up to be an NP-hard. Finding the solution is NP-hard, because finding the optimum entails running our algorithm for all the combinations of size k from a set of size n . This is a combinatorial problem that becomes intractable even for small sizes of k or n (e.g., $k = 5$, $n = 54$). For instance, if we would like to find the optimum bandwidth for a cluster of size $k=3$, then the algorithm need to run for every possible (non repeating) combination of size 3 from the set of size 54. That is for 54 nodes we would end up having 25K combinations ($choose(54,3)$), or 25K possible nodes to start with. We managed to do this and the optimum average bandwidth obtained was 62.7 Mbps. The optimum bandwidth obtained for $k = 2$ was 49.1 Mbps, and for $k = 1$ was 16.9 Mbps. However the computation time took very long (65 hours for $k = 3$, 30 minutes for $k = 2$ etc.), comparing to BASP where it took 23 seconds for $k = 3$ and 15 seconds for $k = 2$. Table 1 shows the BASP improvement over Naive K-Means algorithm. Furthermore, Table 1 shows some centrality measures and some graph properties obtained for each cluster head. To summarize, BASP is able to achieve good bandwidth performance with very low computation complexity.

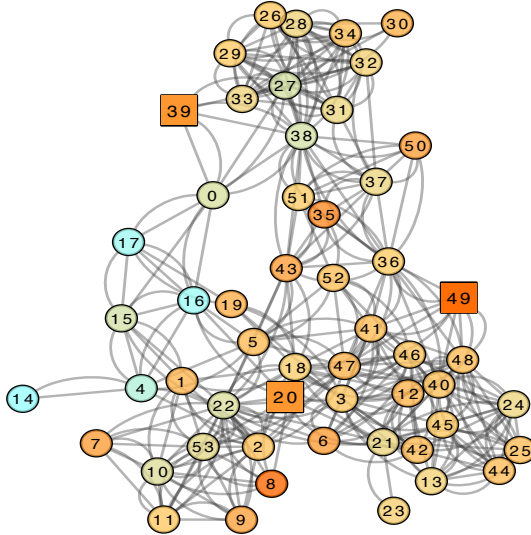


Figure 9: Neighborhood Connectivity

Correlation with centrality metrics: Figure 9 shows the neigh-

borhood connectivity graph of the QMP network. The neighborhood connectivity of a node n is defined as the average connectivity of all neighbors of n . In the figure, nodes with low neighborhood connectivity values are depicted with bright colors and high values with dark colors. It is interesting to note that the nodes with the highest neighborhood connectivity are the cluster heads obtained with our BASP algorithm. The cluster heads (for $k=2$ and $k=3$) are illustrated with a rectangle in the graph. A deeper investigation into the relationship between service placement and network topological properties is out of the scope of this paper and will be reserved as our future work.

5. RELATED WORK

Service placement is a key function of cloud management systems. Typically, by monitoring all the physical and virtual resources on a system, it aims to balance load through the allocation, migration and replication of tasks.

Data centers: Choreo [10] is a measurement-based method for placing applications in the cloud infrastructures to minimize an objective function such as application completion time. Choreo makes fast measurements of cloud networks using packet trains as well as other methods, profiles application network demands using a machine-learning algorithm, and places applications using a greedy heuristic, which in practice is much more efficient than finding an optimal solution. In [8] the authors proposed an optimal allocation solution for ambient intelligence environments using tasks replication to avoid network performance degradation. Volley [3] is a system that performs automatic data placement across geographically distributed datacenters of Microsoft. Volley analyzes the logs or requests using an iterative optimization algorithm based on data access patterns and client locations, and outputs migration recommendations back to the cloud service.

Distributed Clouds: There are few works that provides service placement in distributed clouds with network-aware capabilities. The work in [15] proposes efficient algorithms for the placement of services in distributed cloud environment. Their algorithms need input on the status of the network, computational resources and data resources which are matched to application requirements. In [9] authors propose a selection algorithm to allocate resources for service-oriented applications and the work in [4] focuses on resource allocation in distributed small datacenters.

Service Migration: Regarding the service migration in distributed clouds, few works came out recently. The authors in [20] and [19] study the dynamic service migration problem in mobile edge-clouds that host cloud-based services at the network edge. They formulate a sequential decision making problem for service migration using the framework of Markov Decision Process (MDP) and illustrate the effectiveness of their approach by simulation using real-world mobility traces of taxis in San Francisco. The work in [16] studies when services should be migrated in response to user mobility and demand variation.

While our focus in this paper is to design a low-complexity service placement heuristic for community network clouds to maximise bandwidth, another closely related work is [17] which proposed several algorithms that minimize the coordination and overlay cost along a network.

6. CONCLUSION

In this paper, we first motivated the need for bandwidth-aware service placement on community network micro-cloud infrastructures. Community networks provide a perfect scenario to deploy and use community services in contributory manner. Much previ-

Table 1: Centrality measures for cluster heads

| | k=1 | k=2 | | k=3 | | | k=5 | | | | |
|---------------------------------------|---------|---------|---------|------------|------------|-------------|---------|--------|---------|---------|---------|
| Clusters [node id] | C1 [27] | C1 [20] | C2 [39] | C1 [20] | C2 [39] | C3 [49] | C1 [20] | C2 [4] | C3 [49] | C4 [51] | C5 [39] |
| Head degree | 20 | 6 | 6 | 6 | 6 | 10 | 6 | 10 | 10 | 12 | 6 |
| Neighborhood Connectivity | 7.7 | 9.6 | 9.6 | 9.6 | 9.6 | 10.8 | 9.6 | 8.7 | 10.8 | 8.1 | 9.6 |
| Diameter | 6 | 5 | 3 | 4 | 3 | 5 | 4 | 2 | 3 | 1 | 3 |
| Naive K-Means Bandwidth [Mbps] | 16.6 | 18.3 | | 23 | | | 23.4 | | | | |
| BASP Bandwidth [Mbps] | 16.9 | 27.7 | | 32.9 | | | 38.5 | | | | |
| BASP Running Time | 7 sec | 15 sec | | 23 sec | | | 30 sec | | | | |

ous work done in CNs has focused on better ways to design the network to avoid hot spots and bottlenecks. As services become more network-intensive, they can become bottle-necked by the network, even in well-provisioned clouds. The case in community network clouds is even more hair-raising, with limited capacity of nodes and links and an unpredictable network performance. Without a network aware system for placing services, poor paths can be chosen while faster, more reliable paths go unused.

Furthermore, we proposed a low-complexity service placement heuristic called BASP to maximise the bandwidth allocation in deploying a CNMC. We presented algorithmic details, analysed its complexity, and carefully evaluated its performance with realistic settings. Our preliminary results show that BASP consistently outperforms the currently adopted random placement in Guifi.net by 35%. Moreover, as the number of services increases, the gain tends to increase accordingly.

As a future work, we plan to deploy our service placement algorithm in a real network segment of Guifi.net, using real services and quantify the performance and effects of the algorithm.

7. REFERENCES

- [1] Guifi.net: Services of Catalunya (by zone). <https://guifi.net/en/node/2413/view/services>.
- [2] Stirling Number of the Second Kind. <http://mathworld.wolfram.com/StirlingNumberoftheSecondKind.html>.
- [3] S. Agarwal et al. Volley: Automated data placement for geo-distributed cloud services. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 2–2, Berkeley, CA, USA, 2010. USENIX Association.
- [4] M. Alicherry and T. Lakshman. Network aware resource allocation in distributed clouds. In *Proceedings of INFOCOM, IEEE*, pages 963–971, March 2012.
- [5] B. Braem et al. A case for research with and on community networks. *ACM SIGCOMM Computer Communication Review*, 43(3):68–73, July 2013.
- [6] L. Cerdà-Alabern, A. Neumann, and P. Eschrich. Experimental evaluation of a wireless community mesh network. In *Proceedings of the 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '13, pages 23–30, New York, NY, USA, 2013. ACM.
- [7] D. V. et al. A technological overview of the guifi.net community network. *Computer Networks*, 93, Part 2:260 – 278, 2015. Community Networks.
- [8] K. Herrmann. Self-organized service placement in ambient intelligence environments. *ACM Trans. Auton. Adapt. Syst.*, 5(2):6:1–6:39, May 2010.
- [9] A. Klein, F. Ishikawa, and S. Honiden. Towards network-aware service composition in the cloud. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 959–968, New York, NY, USA, 2012. ACM.
- [10] K. e. a. LaCurts. Choreo: Network-aware task placement for cloud applications. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 191–204, New York, NY, USA, 2013. ACM.
- [11] A. Lertsinsruttavee, L. Wang, A. Sathiaselan, J. Crowcroft, N. Weshsuwannarugs, A. Tunpan, and K. Kanchanasut. Understanding internet usage and network locality in a rural community wireless mesh network. In *Proceedings of the Asian Internet Engineering Conference*, AINTEC '15, pages 17–24, New York, NY, USA, 2015. ACM.
- [12] A. Neumann, E. Lopez, and L. Navarro. An evaluation of bmx6 for community wireless networks. In *8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2012 I, pages 651–658, Oct 2012.
- [13] M. Selimi et al. Cloud services in the guifi.net community network. *Computer Networks*, 93, Part 2:373 – 388, 2015. Community Networks.
- [14] M. Selimi et al. Integration of an assisted p2p live streaming service in community network clouds. In *Proceedings of the IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom 2015)*. IEEE, Nov. 2015.
- [15] M. Steiner and B. e. a. Gaglianella. Network-aware service placement in a distributed cloud environment. In *Proceedings of the ACM SIGCOMM 2012 Conference*, SIGCOMM '12, pages 73–74, New York, NY, USA, 2012. ACM.
- [16] R. Urgaonkar et al. Dynamic service migration and workload scheduling in edge-clouds. *Performance Evaluation*, 91:205 – 228, 2015. Special Issue: Performance 2015.
- [17] D. Vega, R. Meseguer, G. Cabrera, and J. Marques. Exploring local service allocation in community networks. In *10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'14)*, IEEE, pages 273–280, Oct 2014.
- [18] L. Wang, S. Bayhan, J. Ott, J. Kangasharju, A. Sathiaselan, and J. Crowcroft. Pro-diluvian: Understanding scoped-flooding for content discovery in information-centric networking. In *Proceedings of the 2nd International Conference on Information-Centric Networking*, ICN '15, pages 9–18, New York, NY, USA, 2015. ACM.
- [19] S. Wang et al. Dynamic service placement for mobile micro-clouds with predicted future costs. In *IEEE International Conference on Communications (ICC)*, pages 5504–5510, June 2015.
- [20] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung. Dynamic service migration in mobile edge-clouds. *CoRR*, abs/1506.05261, 2015.